ショートノート

パターン抽出によるビットデータ圧縮法†

河村 知行††

テキスト用データ圧縮法 PEM（自動的なパターンの抽出によるデータ圧縮法）を拡張した，ビットデータ圧縮法（VPEM, D2PEM）について述べている．VPEM を種々の画像データに対して適用した結果，かなり良い圧縮結果を得た．特に，タイリングやハッチングによる塗り絵的な画像データや，それらの混合画像に対して非常に良い圧縮結果を得た．

1. パターン抽出法によるデータ圧縮の原理

本実験で使用したデータ圧縮法は，パターン抽出法（Pattern Extraction Method PEM）[1] によるデータ圧縮法の変形である．本節では，テキスト用のデータ圧縮法である PEM の概略について述べる．

PEM は与えられたテキストから PEM 木と呼ばれる探索用のリスト構造を作って，新しい圧縮コードに置き換えるべきパターン（文字列）を決定する．テキスト中のそのパターンをすべて新しい圧縮コードに置き換えて，その置き換えたテキストの後に，そのパターンを付け加える．さらに各パターンを区切るために，区切りコード（〈128〉または $ で示す）をテキストの最後に付け加える．〈128〉は 128 を文字コードとするバイトデータを示す．以上の過程を，置き換えるべきパターンが無くなるまで繰り返すことにより最終圧縮テキストを得る．簡単なテキストが，圧縮テキストになる過程を**図 1** に示す．矢印が新しい圧縮コードへの置換を示している．図 1 には現れていないが，圧縮コードがパターンの一部になることも許されている．

2. PEM アルゴリズムの可変ビット長への拡張

PEM はテキストデータ用のアルゴリズムであるため，その処理はすべてバイト単位（8 ビット）で行われる．このバイト単位の処理の制限をはずした PEM が，可変ビット長 PEM（以後，VPEM と呼ぶ）である．VPEM では元データの要素を n ビットとするとき，2 の n 乗を区切りコードとし，その後の値を圧縮コードとしている．以後，元データの要素と区切りコードと圧縮コードを合わせて，単位要素と呼ぶ．

VPEM では，**図 2**（a）のようなビット列が図 2（b）のようなビット列に圧縮される．図 2 は，元データ要素 1 ビット・圧縮コード 2 ビットの場合である．矢印が圧縮コードへの置換を示している．図 2（b）の先頭の 4 ビット "0010" は，単位要素のビット長が 2（10 進）であることを表している．その後に続くビットは，2 ビットずつで意味を持つことになる．"11" は圧縮コードである．"00" と "01" は元データ要素の "0" と "1" を表している．"10" は区切りコードである．この例では，41 ビットのデータを 38 ビットに圧縮しており，圧縮率は 0.927 ということになる．

以上のアルゴリズムでは，単位要素のビット長が変わるところ（例えば，圧縮コードが 15 から 16 に変わるところ）で，一時的に圧縮率が大きくなってしまう．そのために，圧縮コードが大きくなるに従って圧縮率が波打つように変化するので，どの圧縮コードで圧縮を止めるかが問題となる．今回の実験では，置き換えるべきパターンが無くなるまで圧縮を行い，その中で最小の圧縮率を結果としている．しかし，実験の結果からほとんどの場合，圧縮コードが 255 または 511 までの間に最小の圧縮率が出現することがわかったので，実用上の問題はないと思われる．

3. 二次元に拡張された VPEM

VPEM の拡張として，圧縮コードに置き換えられるべきパターンとして二次元的なパターンを考えることが可能である．これを D2PEM と呼ぶことにする．

VPEM は，一次元的なテキストを圧縮の対象とし

† Bit-Data Compression Method by Pattern Extraction by TOMOYUKI KAWAMURA (Department of Information Electronics, Tokuyama Technical College).
†† 徳山工業高等専門学校情報電子工学科

ていたので，パターンを図3(a)のように成長させていた．一方，D2PEMではパターンを図3(b)のように二次元的に成長させる．

図4(b)は，図4(a)をD2PEMにより圧縮したようすを表している．図4の中の数は10進数であり，実際には，画素2ビット・単位要素3ビットとなっている（2次元処理では元データの要素を画素と呼ぶ）．数4は区切りコードを表している．図4(b)の中の＊印の付いた圧縮コードが右下に広がる正方形のパターンの基点となっており，ファイルに格納するときは，＊印の付いた圧縮コードと区切りコードと画素だけが格納されることになる．図4では，その数は40となる．

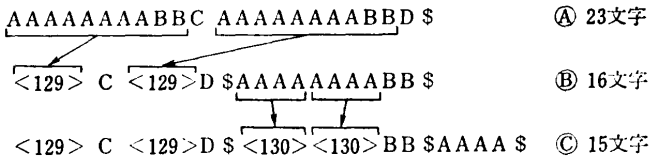

図1 PEMによる圧縮の過程（文献1)より）
Fig. 1 Process of compression by PEM.

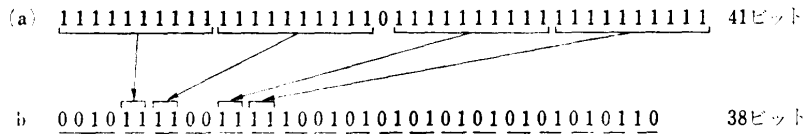

図2 VPEMによる圧縮の例
Fig. 2 Example of compression by VPEM.

図3 二次元パターンの生長の方法
Fig. 3 Growth method of 2 dimensional pattern.

今回の実験では，D2PEMはVPEMと同程度の圧縮率しか得られなかった．より大きな2次元データや，3次元データ（この場合はD3PEMとなる）に対してこのアルゴリズムは適用すべきであろう．

4. 実験と評価

表1に今回の実験結果を示す．HUFF1は，画像データを一次元データとみなしてハフマン符号[2]により圧縮を行うプログラムである．「画素」と「同じ値の画素の連続（ラン）」の両方を合わせてハフマン符号を計算している（2値画像ではランだけで計算している）．ハフマン符号と実際の画素およびランの対応表は，各画像データごとに最良のものを仮定しており，その対応表の大きさは圧縮率の計算に含まれていない．そのため，本当の圧縮率は表1の値より大きくなるはずである．一方，VPEMによる値は，本当の値と言える．なぜならば，VPEMでは，ハフマン符号のような対応表を使用していないからである．HUFF2は，画像データを二次元データとみなして垂直方向の画素の値の差（2値画像では排他的論理和）をとり，その結果を一次元データとみなしてハフマン符号により圧縮を行うプログラムである．

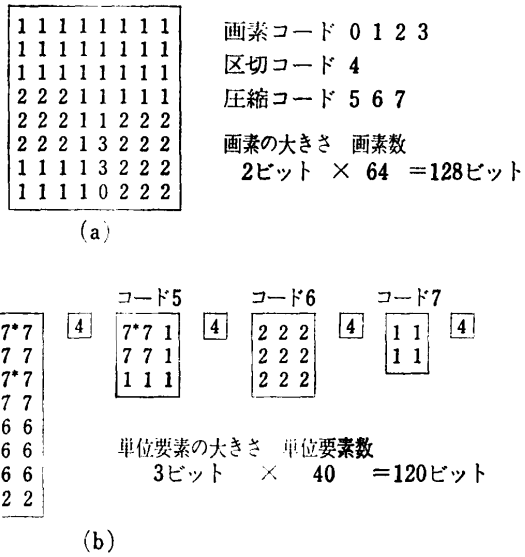

図4 D2PEMによる圧縮の例
Fig. 4 Example of compression by D2PEM.

表1の画素長の欄は，画素のビット長を示している．画素長1は，2値画像データを表し，画素長2, 3, 4は，濃淡の画像データ，またはデジタルRGBの画像データを表している．各画像データは，400×400の画素を持っている．詳細は次のとおりである．

GIRL：女性の顔の濃淡画像である．各画素は8ビ

表 1 VPEM による圧縮率
Table 1 Compression ratio by VPEM.

データ名	画素長	HUFF 1	HUFF 2	VPEM
GIRL	1	0.313	0.273	0.385
GIRL	2	0.398	0.336	0.360
GIRL	3	0.419	0.329	0.378
GIRL	4	0.507	0.370	0.467
DITHER	1	0.857	0.728	0.303
CAPIX	1	1.008	0.999	1.176
SINGLE	1	1.003	0.753	0.725
BUNSHO	1	0.416	0.394	0.465
SLANT	1	1.002	1.003	0.0012
MICKEY	2	0.132	0.136	0.128
MIX	2	0.638	0.655	0.121
HATCH	1	1.007	1.007	0.207
TILE 2	2	1.021	1.117	0.151
TILE 3	3	0.919	0.939	0.103

ットで，その上位数ビットを実験に使用している．

DITHER：上記の GIRL に 4×4 のディザ法を適用して，濃淡を2値表現したものである．

CAPIX：上記の GIRL を CAPIX 法[3]により2値表現したものである．

SINGLE：CAPIX 法を単純化した方法により GIRL を2値表現したものである．CAPIX 法より，“0” や “1” の連続が多い方法である．

BUNSHO：イメージスキャナにより，読み込んだ活字の文書データ（2値）である．

SLANT：斜め 45 度の平行線を 265 本引いた，単純な幾何学的な2値画像である．

MICKEY：ミッキーマウスの塗り絵（4色）である．

MIX：MICKEY の背景色のみを SLASH で塗り換えた，塗り絵とハッチングの混合画像である．

HATCH：MICKEY の4色をそれぞれ特定のハッチングパターンで塗り換えたものである．

TILE 2：MICKEY の4色をそれぞれ特定のタイリングパターン（4色使用）で塗り換えたものである．

TILE 3：MICKEY の4色をそれぞれ特定のタイリングパターン（8色使用）で塗り換えたものである．

表1からもわかるように，画像データについて VPEM が良い圧縮率を示している．特に，ハッチングやタイリングを用いた画像データに対して非常に良い圧縮率を示している．また，その他の画像データに対しても，ハフマン符号と同程度の圧縮率を示している．このことから，VPEM が汎用的な画像データ圧縮法として有効であることがわかる．さらに，MICKEY と MIX との比較により，混合画像データに対しても有効であることがわかる．このように，VPEM は汎用性が高く，画像と文字を単一のデータ圧縮法 (VPEM) により圧縮できるという長所を持っている．

HUFF 1・2 で1を超えている値があるのは，ランが短すぎるために，ランに与えたビット分が圧縮率に悪影響を及ぼしているためである．

5. まとめ

テキスト用データ圧縮法である PEM を拡張した，ビットデータ用圧縮法について報告した．実験データとしては画像データを用いたが，他のデータに付いても適用可能であると考えられる．PEM も VPEM も多くの計算を必要とするアルゴリズムであるので，専用のハードウェア[4]の実現が望まれるところである．

参考文献

1) 河村知行：自動的なパターン抽出によるデータ圧縮法の提案，情報処理学会論文誌，Vol. 25, No. 6, pp. 1089-1094 (1984).
2) 宮川 洋，原島 博，今井秀樹：情報と符号の理論，p. 266，岩波書店，東京 (1982).
3) 土屋博義，中里克雄：階調画像の2値再生法，昭和60年度電子通信学会総合全国大会論文集，p. 5-212 (1985).
4) Kawamura, T.: Data Compression by Hardware PEM (Pattern Extraction Method) Using Multi Processor Elements, *J. Inf. Process.*, Vol. 9, No. 4, pp. 213-219 (1987).

(昭和61年10月27日受付)
(昭和62年7月9日採録)

河村 知行（正会員）

昭和28年生．昭和51年東京教育大学理学部応用数理学科卒業．昭和54年筑波大学大学院博士課程数学研究科中途退学．修士修得．昭和54年徳山工業高等専門学校情報電子工学科勤務，現在に至る．計算機システム，計算機アーキテクチャ，アルゴリズム，マンマシンインタフェースなどに興味をもつ．